Rapid Development of Intelligent Vehicle Highway System Projects

by

William H. Spuck
Robert L. Phen
Ronald C. Heft

ABSTRACT

The Jet Propulsion Laboratory has pioneered an approach for the management of complex projects. The approach is called the Rapid Development Method (RDM). RDM suggests an attractive paradigm for the development of Intelligent Vehicle Highway System (IVHS) concepts, systems, components, and demonstrations up to and including the development of the commercial infrastructure needed for large-scale deployment.

RDM was innovated to address directly the well known problems of conventional technical project methods, namely, cost overruns, schedule delays, equipment obsolescence by the time it is fielded, general user dissatisfaction with the resulting system, and inflexibility to technical and programmatic changes. RDM addresses these through a sequence of frequent incremental deliveries, requirements feedback from development and operational experience, extensive user interaction during development, and progressive formality of the development process.

IVHS applications represent the kind of complex systems that can benefit from the planned evolutionary infusion of progressively better technology coupled with real-world evaluation by typical highway system users. This paper discusses the potential of this relationship between RDM and IVHS.

INTRODUCTION

Intelligent Vchiclcs Highway System (IVHS) projects have, in general, the following characteristics

o They arc evolutionary and can bc implemented incrementally.

e They include feedback to the requirements which arc revi sed, based on the work accompli ished.

● They include involvement of the users and stakeholders.

● They include increasing formality, e.g., regulations anti standards will evolve in time.

Convent ional government acquisition methods assume characteristics which, in fact, arc myths. These assumptions arc:

⊙ Requirements can bc set 01 frozen at the beginning of a project.

⊙ Budgets and funding profiles arc fixed.

⊙ Technology will not change.

⊙ Users can wait three to eight years for a system to bc delivered.

These assumptions cannot bc made for IVHS projects.

Conventional government procurement methods, then, become less than satisfactory for implementing IVHS projects. We believe that the Rapid Development Method is a preferred approach for IVHS project applications.

The Rapid Development Method (R])M) is a project management approach pioneered at the Jet Propulsion Laboratory (JPL) and aimed at solving problems frequentl y encountered during system acquisition and development. These problems arc wide] y known to include cost growth, long development cycles, program delays, inflexibility to inevitable program matic and technical changes, obsolescence of system elements before fielding, and general user dissatisfaction with the fielded system. RDM addresses these problems through a combination of procedural, relational, and philosophical innovations.

This paper introduces RDM through a (discussion of its tenets and its contrasts with the Conventional Development Methodology (CDM). [CDM is assumed by governmental acquisition methods.] Then, the application of RDM to IVHS is illustrated by example.

RAPID DEVELOPMENT

RDM is an outgrowth of software rapid prototyping techniques, but it is not rapid prototyping. Rapid prototyping is practiced to validate requirements and design concepts. The intent of a prototype is to try out new ideas; more specifically, it evaluates new or alternative sets of requirements. When a prototype's results arc known, it generally is abandoned. On rare occasions it is reworked into an operational system, but in general it lacks the implementation rigor and]onc-term sustainability needed by Operational systems.

RDM, in contrast to rapid prototyping, is practiced in order to implement systems. There is no intent to throw anything away (though, from time to time, system elements may be replaced as obsolete). ‘J-hc intent is to use the implemented system operationally after each increment is delivered. Under RDM it is expected that require.mcnts will change in order to accommodate new understandings of the problem and new technology. 1 ‘last of change is a cornerstone. of the RDM concept; change is welcomed to improve the system. Rapid development and rapid prototyping arc compared in Figure 1. Both concepts arc important to IVHS since it is expected that IVHS will implement both demonstrations and early deployment corridors.

Rapid development is a specific project management approach. It is described by a specific set of tenets. It defines its own approaches to most real system implementation issues, For example, RDM has a formalized systems engineering procedure, a specific approach to configuration management, anti rigorous documentation procedures. All of these approaches are traceable to at least the intent of the corresponding items for CDM, but they are tailored to respond to the tenets and pace of RDM.

THE FOUR TENETS OF RDM

RDM has four basic tenets:

(1) Incremental 1 lc.liveries.

(2) Requirements Feedback.

(3) 1 Extensive User Interaction.

(4) Progressive Formality.

1 lath is defined and hen discussed n this section.

Incremental Delive ies

The first te.net states that RDM involves a series of incremental deliveries. Each delivery constitutes an operable, functionally valuable, partial system. The overall system is developed and delivered to its users (and thereby contractually delivered to its sponsor) in small evolutionary increments. The users employ the evolving system in the daily conduct of their mission,

This contrasts sharply with the conventional development method (CDM), an approach that involves a single, all-inclusive delivei y and is often called "big bang. " CDM is a process which takes three to eight years to execute. Only at the end of this period does it deliver its product, the system, to the users.

Figure 2 depicts the schedule of a typical CDM project. 1 rom the overall appearance of this figure, it is clear why this life'cycle is often called a "waterfall. "

· CDM phases are implemented strictly sequentially. In fact, development cannot pass into the next phase unt i 1 all aspects of previous work arc completed, reviewed, and approved. Specifically, requirements arc not developed until planning is complete; design dots not begin until the requirements arc understood (often called validated); implementation of the design dots not occur until the design has been reviewed formally (often called a Critical Design Review); integration and test occur in a formal way after some kind of test readiness review shows that all components have been implemented properl y; and, of course, nothing is installed until it has been thoroughly bought off by a testing (verification) program. The system is certified after installation and, finally, the system is placed into operation after a formal Transfer- to- Operat ions evaluation. Once in the user's hands, of course, the system must be sustained.

This sequential nature explains immediately why CDM so frequently encounters delivery delays. System elements completed early in any life cycle phase must wait for lagging system elements before all proceed together into the next phase. All elements wait for the latest element. Prolonged schedules arc inevitable. The concomitant inefficiencies in staff utilization arc also a key cause of CDM cost growth.

It is important that the phases of CDM not be confused with the incremental deliveries of RDM.

Figure 3 presents an RDM life cycle. The project, first of all, is typified by a series of deliveries. 1 Lath delivery must go through a "mini" life cycle of its own consisting of "mini" phases. These echo the phases of the conventional life cycle. They arc detectable in Figure 3, where they arc represented by repeat ing the phase numbers provided in Figure 2. The number 1 refers to planning, 2 to requirements, and so forth. As shown on the chart, phases 2 through 7 arc conducted once for each dc] ivery. Of course, each delivery must be sustained until the next delivery is complete. Once a delivery has been transferred to operations, the previous delivery has been superseded. It (disappears, and the new delivery is the one to repeatedly be sustained.

It is important to note that the RDM delivery interval, the time between transfers of capability to user operations, must be fairly short. JPL experience has shown that a delivery interval of more than about 12 months begins to 1osc the characteristics of RDM and gather the characteristics of CDM. On the other hand, it also has proven risky to sustain a series of RDM deliveries at intervals of six months or less. The inevitable fixes which must occur to any system immediately after cic.livery then consume too much of the project team's attention during a short delivery life cycle.) ixperience suggests that a delivery interval of approximately nine months is optimum. It is short enough to insulate a delivery to the first order from funding and requirements volatility; it achieves the tenets and objectives of RDM; but at the same time, it offers enough time to conduct an effective cle.livery life cycle.

Requirements Feedback

The second tc.net is that RDM expects active feedback from the experience gained from onc incremental delivery to the. requirements for the next.

A perspective from which to investigate the feedback aspects of RDM is gained by contrasting the R])M and CDM engine.cxin~ processes. Figure 4 diagrams the CDM engineering process, greatly simplified, and crafted to emphasize. contrasts with RDM. An impel-tant point to note is that the work flow is linear. As depicted in 1 igure 2 and again here, work is done first on requirements, then on design, then on careful, detailed planning, followed by implementation (which includes detailed design and the implementation of the hardware, software, and "peopleware" that goes into a systcm), and final] y cm testing. The substeps of implementation arc not elaborated in this figure; RDM is not significantly different from CDM at this point (though its schedule is significantly different).

Contrasting with this CDM work flow is Figure 5, the RDM engineering process. As shown by the work flow arrow at the very top of the figure, RDM workflow is not line-ar. Work at first proceeds linearly through phases, but then encounters two major differences. First, once the first increment is delivered, the work flow process is reexecuted for the next delivery. RDM involves a cyclical procedure. Second, only a fraction of the requirements are selected for implementation at any delivery. The capabilities represented by these arc added to the capabilities of earlier deliveries, as illustrate.d in Figure 6. RDM, then, delivers a sequence of system upgrades. The value of the system increases incrementally as shown in Figure '7. This is in contrast with value under CDM, which is only realized at final delivery.

As RDM periodicall y delivers to the users an increment of capability, the users arc able to provide understanding of how effectively that dc.livery is meeting their needs. As the users assess the impact of a delivery on their operations, the system developer is able to work with the users to adjust the system require-mc.nts to better satisfy operational needs. That adjusted set of requirements becomes the basis for all subsequent incremental deliveries. This feedback process is formal and proactive. 1 t is a key element in making RDM effective from a user's perspective.

Extensive User Interaction

The third tenet is extensive user interaction during development. Interaction refers not on] y to the feedback of requirements from one delivery to future deliveries, but also the intimate involvement of the users during the implementation life cycle of each delivery. RDM embraces the. premise that the more the real users arc involved, the more effectively the system will m'cet their needs. 'J'bus, the RDM process includes a role for users in virtually every step of the delivery life cycle and involves them, at a minimum, in the key decision processes leading to each delivery. Figure 8 lists the places in which RDM involves users, showing that users are intimately involved throughout -- from requirements to design to test to certification to feedback on the system's success.

Progressive Formality

Finally, thc fourth defining tenet of RDM is progressive formality. Under RDM, the first delivery will be done quite quickly and with very little formality, much like a rapid prototype, As succeeding deliveries arc undertaken, implementation procedures become more formal and more comprehensive. Procedures and products that under CDM would have to be dcmc perfectl y before the next step in the i mplementation life cycle can begin arc done for RDM under a planned progression of thoroughness, so that at the final delivery they converge to the same degree of formality they would have achieved if the system development had been under CDM.

Table 1 illustrates progressive formality by tracing the formality of early, middle, and late deliveries in the domain of several engineering processes and products. For example, documentation for determining dc.livery goals (requirements documents) and for helping users operate the system is emphasized for early deliveries. Middle deliveries add an emphasis on documents that help system administrators control the system. By the final delivery, emphasis shifts [0 assuring that all required documents arc complete and of high quality. Of course, throughout the succession of deliveries, attention is given to capturing information as it becomes available. It makes little sense to set information aside in informal documents when the final formal documents arc available and evolving.

Formality all along is aided by the limited scope of each delivery. Formality under CDM is often resource consuming simply because the scope and complexity of a single delivery make the processes daunting.

RDM AND ‘J’] IE MYTIIS OF CDM

The CDM life cycle is based on a, set of assumptions. Some of these assumptions arc patently true. Others arc declared to be true in order to provide an operating basis for the methodology, but in practice they arc, at best, forcibly maintained to enable the methodolog y to succeed. Wc cal l hese latter assumptions the '{myths" of CDM. Key among them are:

Myth 1 - Requirements can be set and frozen at the beginning of a project.

Myth 2 - Budgets and funding profiles can be set and frozen at the beginning of a project and realized as the project unfolds.

Myth 3 - Interfacing systems will not evolve during the life of the project.

Myth 4 - Technology will not change dramatically during the life of the project.

Myth 5 - Users understand it will take three to eight years to deliver a system.

To anyone who has been involved in project work and used the CDM it is clear that the above are indeed myths.

It should be clear that perfect require.mcnts cannot be stated up front. RDM addresses the myth of frozen requirements by offering a process that allows -- indeed encourages -- requirements i mprovements prior to initiation of each of its incremental deliveries.

Planned budget profiles arc rarely realized. RDM adapts easily to changing programmatic environments, including budget changes, by its ability to rescope each delivery.

The system environment, that is, the elements of other systems with which the system interacts, will not be constant, but will change in time. Changes in system environment arc as easil y accommodated under RDM as arc changes in requirements or programmatics.

To anyone involved with technology, change is inevitable during the 3-to-8 year life cycle of a typical project. RDM addresses time changes with a short delivery life cycle and the opportunity to introduce worthwhile new technology with each successive de.livery.

The individuals involved with systems that take up to eight years to implement will change. Most users would like to have some kind of response to their requirements while they are still in the position they occupied when they stated their needs. RDM provides users an acceptable wait between deliveries.

WHERE AND WHEN TO APPLY RDM

RDM should be considered when a system architecture or final operational concept can be defined early, and when the system can be decomposed into meaningful phases with some Confidence.

Some indicators that RDM can be applied are

- technology is changing rapidly.
- technology or automation is being inserted into ongoing operations.
- the user needs the capability rapidly.
- automation of manual or untried operations requires feedback on progress.
- implementation entails mostly design and deployment.
- it is desirable to keep the user's and sponsor's interests high.
- better management control is needed.
- there is a need to control and respond to evolving requirements or to changing technical, budget, or programmatic environments.

RDM can be applied to both software and hardware projects and even to studies where requirements are changing rapidly.

Not all system developments can benefit from the RDM process. For example, it cannot be applied to systems that must be delivered complete -- a spacecraft, an automobile. But it can be applied to upgrading automobiles if the manufacturing process has been designed to permit it. It cannot be applied when the intrinsic implementation time of the most basic form of the system is long compared to an RDM delivery interval -- a major construction project, chemical processes with very slow reaction rates. But it might apply if the overall project can be partitioned into subprojects -- i.e., highway construction.

RDM AND IVHS

Intelligent vehicle highway systems almost inherentl y aim at providing a final operational capability incrementally. They are especially suited to applications of RDM. The highway system itself could have been developed using RDM. Similarly, modifications to the highway infrastructure to implement IVHS could benefit by using RDM. The incremental nature involved in demonstrating evolving technology, such as with the Automated Highway System (AI 1S), is custom made for RDM as the management approach. The AHS has an objective, by law, to achieve an operational capability by 1997. That capability can be implemented incrementally; indeed, it is, almost inconceivable that it would be approached conventionall y. Because of the rapidity of implementation required, RDM would seem to be the preferred manage.mcmt approach. RDM is further applicable to the continued developm ent Of the AHS after 1 99'/ when it will be upgraded and new technologi es evaluated.

The AHS is used as an example of the application of RDM. The following arc assumed for purposes of the example.

- The **1997** operational capability will include a fully automated lane (longitudinal and lateral control) with merge in and out capability on a test track with control center operation.
- The test track exists and needs only to be modified.
- The effort is initiated in January 1994.

Referring to Figure 9, it can be seen that the schedule permits a maximum of five deliveries preceded by a planning phase. Two options arc shown (many more exist): The first assumes a three month planning phase, the second a one-year phase. It is necessary that the planning phase bc sufficient to dcfinc the 1997 final operational capability rcquircmcnts and design. The specific dclivcrics for thcsc two cascs arc shown in "J'able 2.

Later deliveries could includc capabilities for:

- Alternative longitudinal and latcral control systems.

e Automated check-in and chcck-out.

- Malfunction mitigation.
- Merging of nonautomated highway traffic.
- Mixcd vchiclc traffic.

The two above options arc very much simplificd and would serve only as a starting point for planning. The real AHS situation could bc significantly different, e.g., a test track may have to bc built, adequate lateral and longitudinal control systems alrcady exist and could bc easily tcstcd on available test facilities, a merge system could rcquirc considerable work, etc. But these arc exactly the unccrtaintics that commend RDM. The dc.live.rics for the case requiring that a test track bc built could bc those shown as option 3 in 'J'able 2.

in all of the above. cam, cach delivery would rcsult in modification of the final operational capability (1 'OC) rcquircmcnts and changc prcconccptions rcgarding subscqucnt dclivcrics. This fccdback would includc uscr rcsponsc to the delivery. Also, inevitable changcs in funding, tcchnology, etc. will changc preconceptions about futurc dclivcrics. All of these changes would fccd into an evolutionary final operational spccification and ultimately the dcployed FOC itsclf.

SUMMARY

In summary, RDM should bc particularly applicable to the dcploymcnt of systcms which can bc decomposed in time. *In addition to* the AHS, thcsc could includc: advanced traffic management systems, traffic control systems, traveler information systems, routc guidancc and navigation systems, transit fleet management systems, transportation management data base systems, incident management systems, etc.

RDM is also applicable to operational test programs such as the Corridor Program where the delivered technology will remain deployed. When the technology is demonstrated by testing it and then removing it, rapid prototyping should be considered as a management approach.

It should also be noted that RDM is not bounded by transportation mode. RDM can be applied to projects of all modes that have characteri stics that permit incremental deliveries. The approach is also applicable to intermodal IVHS projects.

The majority of IVHS, therefore, have some. characteristics that enable RDM tenets to be appli cd. The overall application of RDM will permit a more rapid deployment of IVHS than would be possible under conventional management methods. Figure 10 shows that simulations, prototype implementat ions, incremental implementations and conventional implementations develop from a final operational capability specification and lead to implementation of IVHS. The final operational capability (FOC) specification is changed by feedback from these activities, as well as from the architecture development. The RDM characteristics can underlie this entire process. The common thread is the evolution of the FOC specification by the application of RDM tenets. During the early definition phase, this evolution of the FOC specification is the distillation of the wisdom of IVHS. Once the specification is understood, RDM forces it on the systems being deployed. We believe that RDM will be more effective in the implementation of IVHS than any other proven methodology.

Table 1. Progressive Formality

Process or Project	Early Deliveries	Middle Deliveries	Late Deliveries
Documentation	User Requirements Documents	Requirements, User, and Administrator Documents	All Documents
Testing	operational Tests	Functional and Performance Tests	Requirements Verification Tests
Tracing	Informal	Formal	PCA/FCA
Product Assurance	Good Practice Suggestions	Process Monitoring	Full Inspection
system Performance Measures	Good Engineering Practice	Improvements	Formal Verification
Configuration Management	Requirements, as Built	Formal at System Integration	All Facets Formal

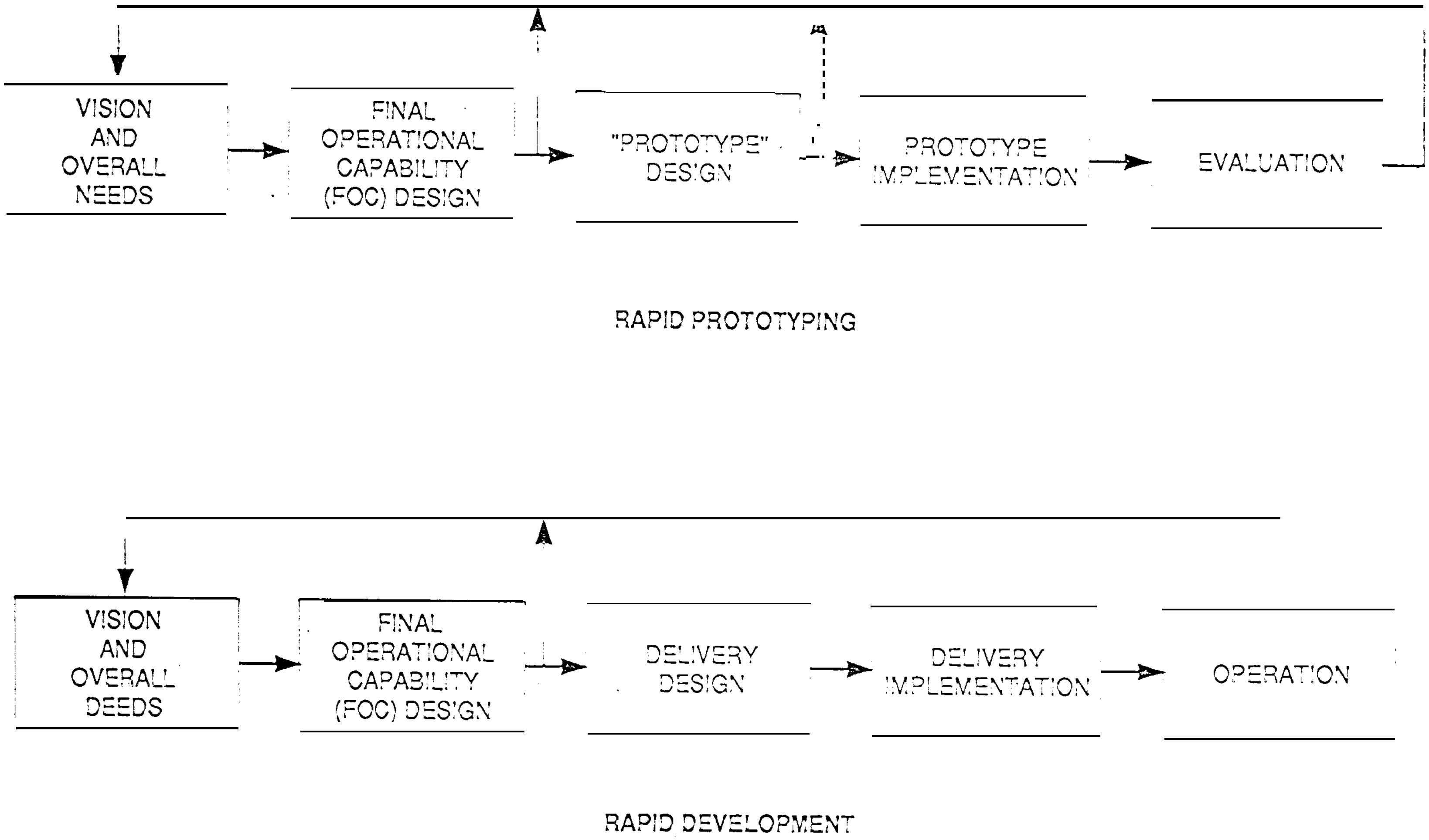
Table 2. RDM Applied to the AHS

1997 OPERATIONAL CAPABILITY
- FULLY AUTOMATED LANE
- MERGE IN AND OUT
- TEST TRACK WITH CONTROL CENTER

	OPTION (1)	OPTION (1)	OPTION (2)
DELIVERY 1	• Test track modified and instrumented	• Test track modified and instrumented • Lateral control complete and tested	• Lateral control complete and tested at available site • Longitudinal control complete and tested at available site
Delivery 2	• Control center complete • Lateral control complete and tested	• Control center complete • Longitudinal control complete and tested	• Combined lateral/longitudinal control complete and tested at available site
Delivery 3	• Longitudinal control complete and tested	• Combined lateral/longitudinal control complete and tested	• Test track and control center complete
Delivery 4	• Combined lateral/longitudinal control complete and tested	• Merge in and out system complete and tested	• Combined lateral/longitudinal control with merge in and out system complete and tested
Delivery 5	• Merge in and out system complete and tested		

(1) Modified existing test track

(2) New test track



RAPID PROTOTYPING

RAPID DEVELOPMENT

Figure 1 of Rapid Prototyping and Rapid Development

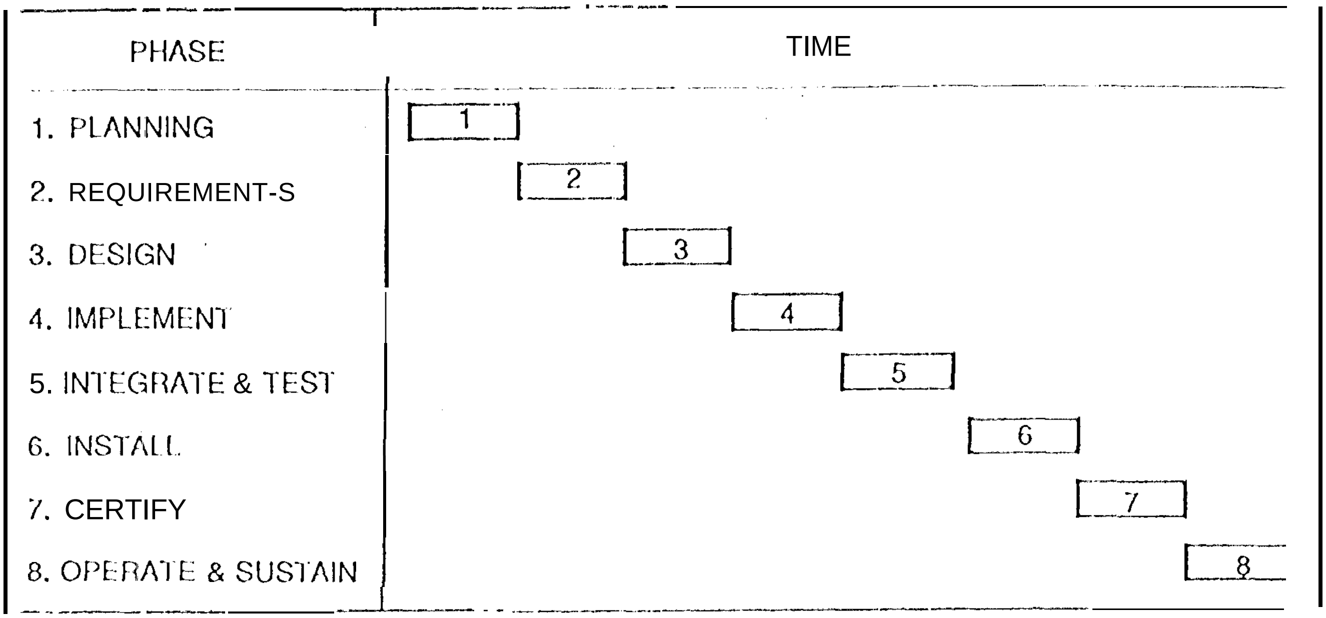
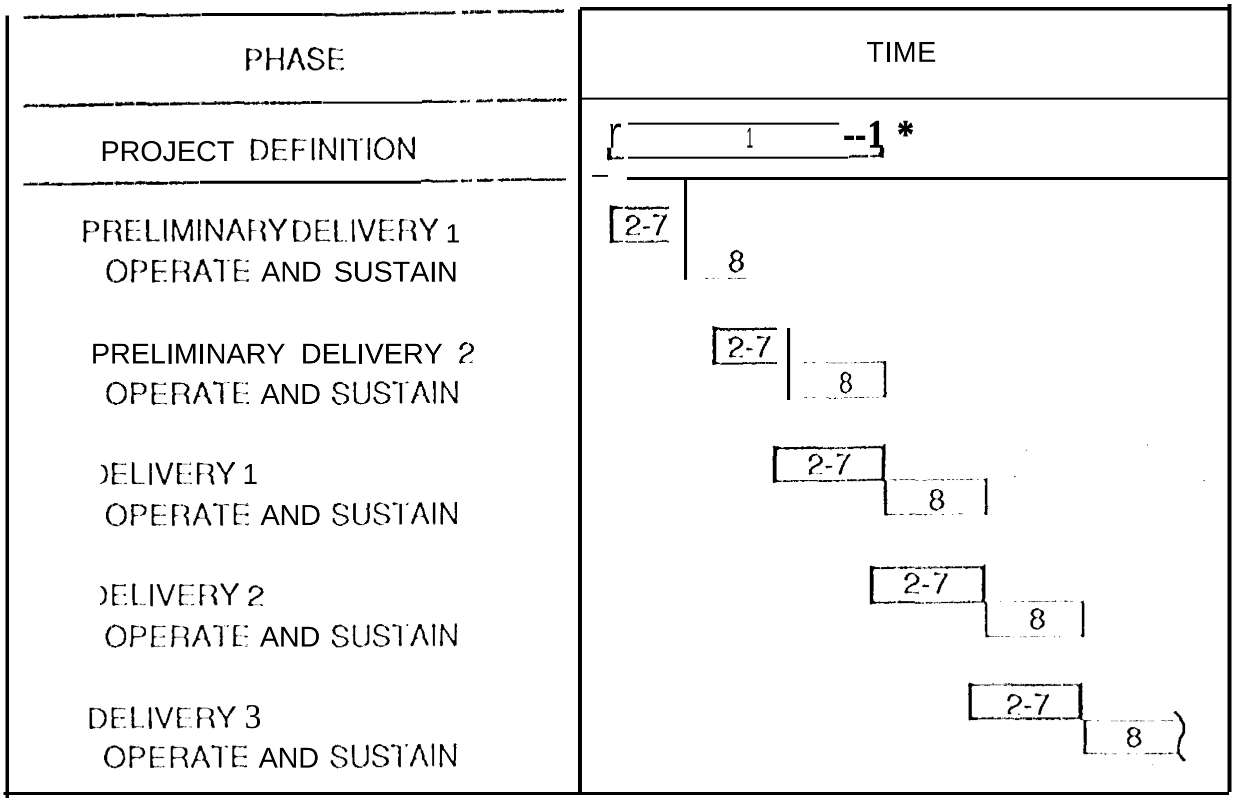
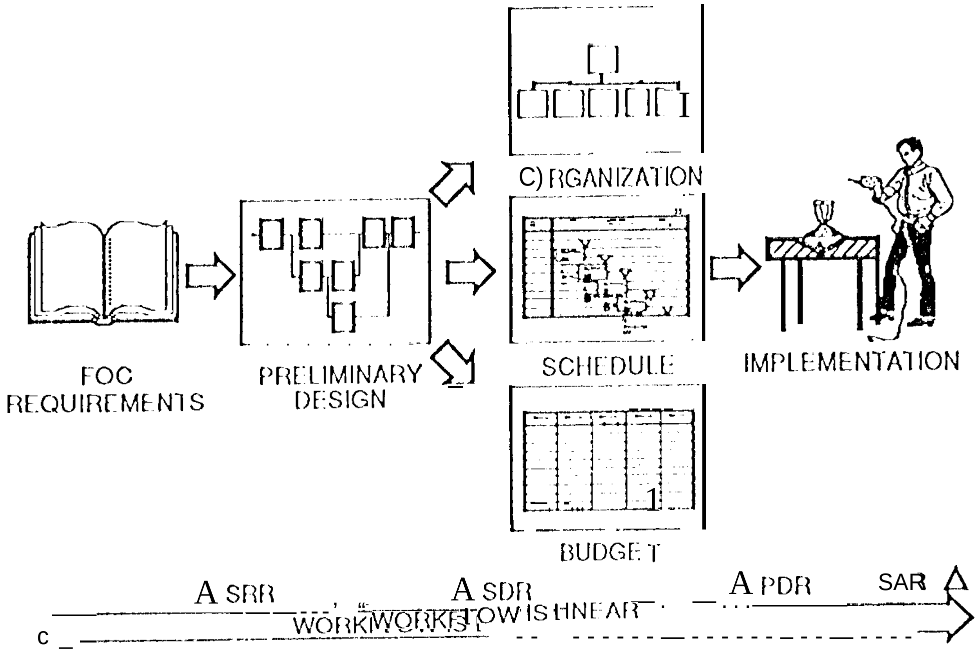


Figure 2. Schedule of a Typical Conventional Development Project



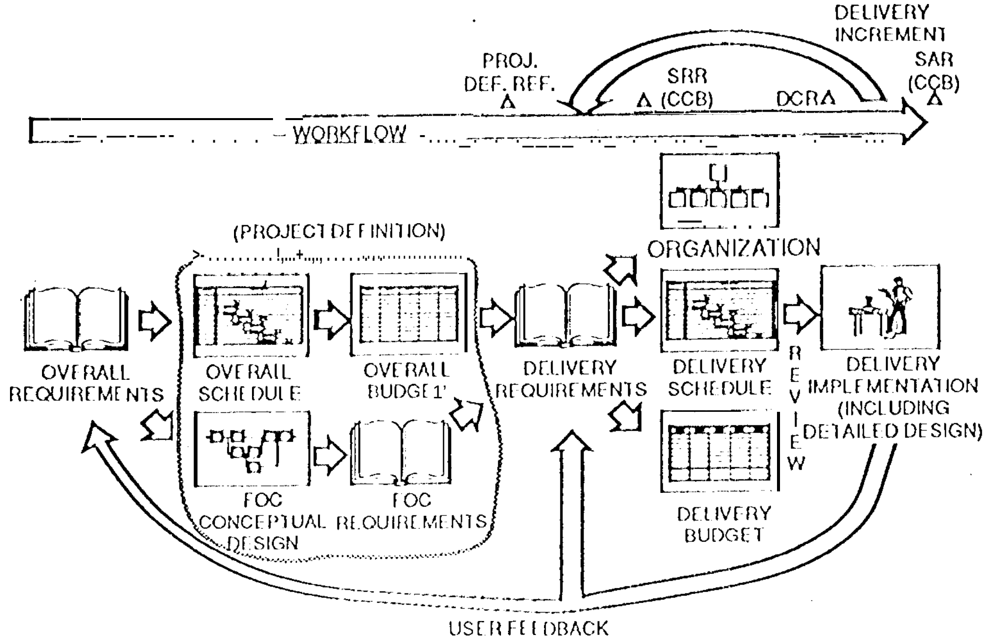
*INCLUDES A LITTLE 2,3

Figure 3. Rapid Development Method Project Schedule



LEGEND:	FOC --	FINAL OPERATIONAL CAPABILITY
	SRR --	SYSTEM REQUIREMENTS REVIEW
	SDR --	SYSTEM DESIGN REVIEW
	PDR --	PRELIMINARY DESIGN REVIEW
	SAR -	SYSTEM ACCEPTANCE REVIEW

Figure 4. Conventional Development Method Engineering Process



LEGEND: FOC - FINAL OPERATIONAL CAPABILITY
SRR -- SYSTEM REQUIREMENTS REVIEW
CCB -- CONFIGURATION CONTROL BOARD
DCR -- DESIGN CRITICAL REVIEW
SAR - SYSTEM ACCEPTANCE REVIEW

Figure 5. Rapid Development Method Engineering Process

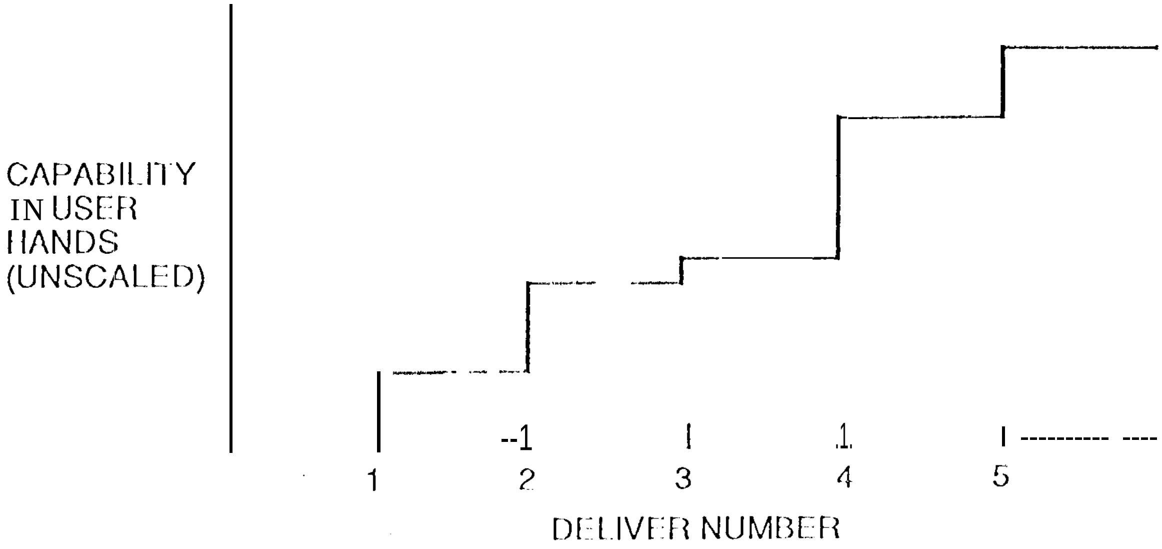


Figure 6. RDM Provides Increasing User Capability with Each Delivery

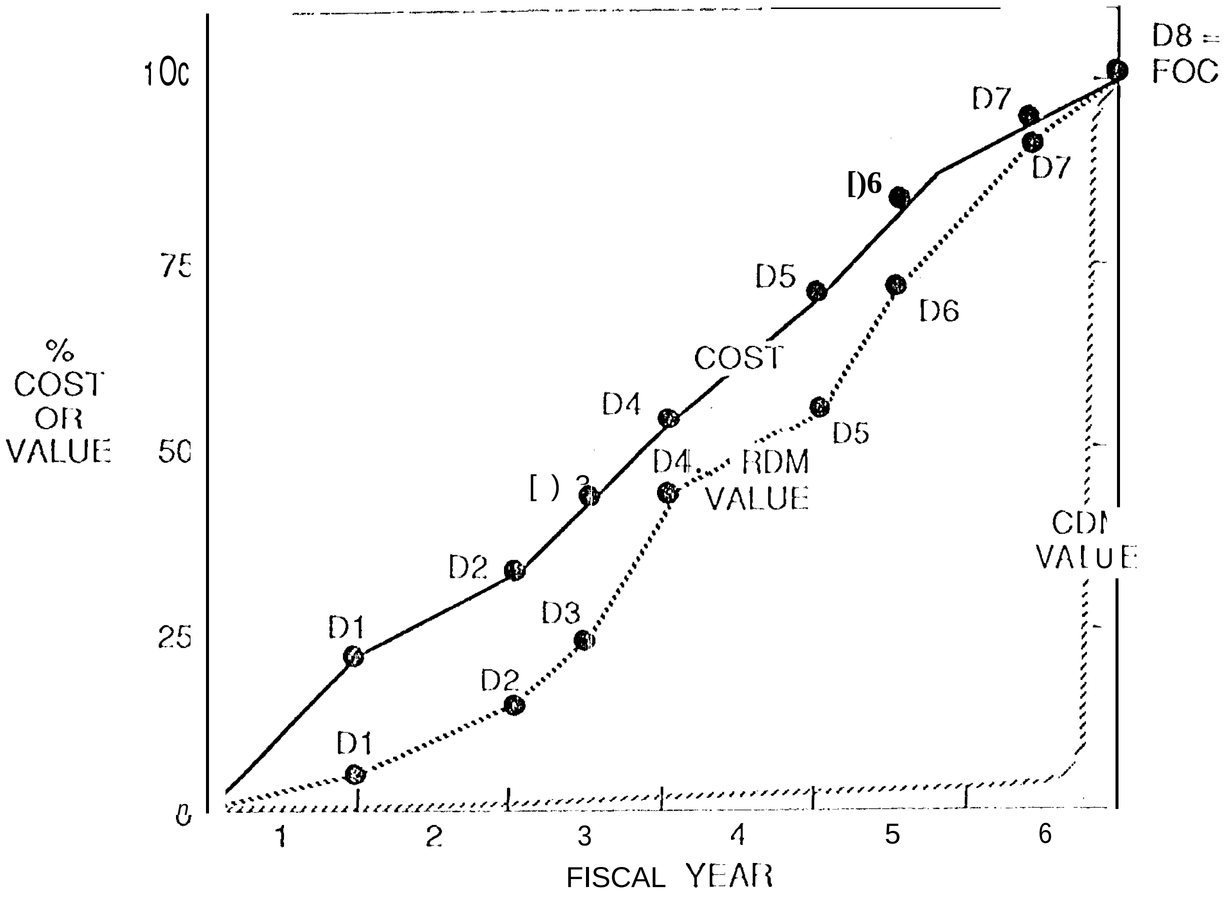


Figure 7. RDM and CDM Value Contrasted

- **USERS** STATE THE REQUIREMENTS UP FRONT
- USERS REVIEW SPECIFICATION AND DESIGN DOCUMENTS
- USERS SET DELIVERY PRIORITIES
- USERS PROVIDE DETAILED REQUIREMENTS TO IMPLEMENTERS

e USERS REVIEW IMPLEMENTATION-IN-PROGRESS

- USERS ASSIST IMPLEMENTERS IN DETAILED COST/CAPABILITY TRADEOFFS
  - WORK IS PA RTITIONED TO INDIVIDUAL USER'S INTERESTS
  - IMPLEMENTATION IS ESSENTIALLY "BUILD TO COST"

e IUSERS TEST I"HE SYSTEM

- USERS ACCEPT THE SYS1 EM
- USERS OPERATE (ARUSE) THE SYSTEM
- USERS CHANGE THE REQUIREMENTS BASED ON OPERATIONAL EXPERIENCE

BENEFIT

USERS ARE INVOLVED!

Figurc 8. User Involvement in RDM

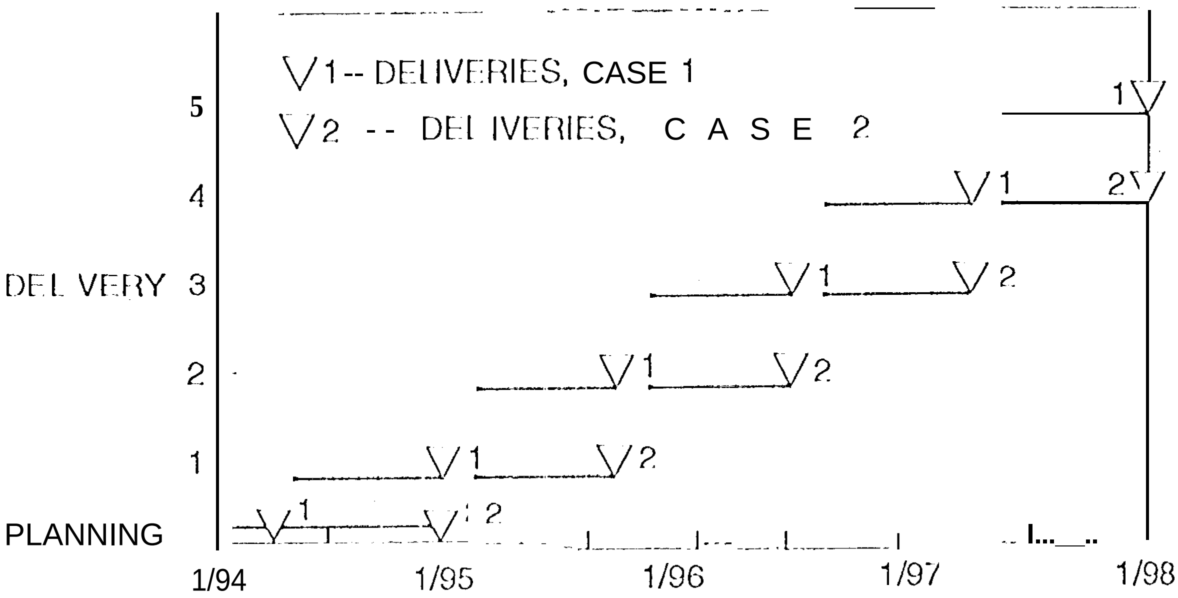


Figure 9. Example 1 Delivery Schedule for AHS

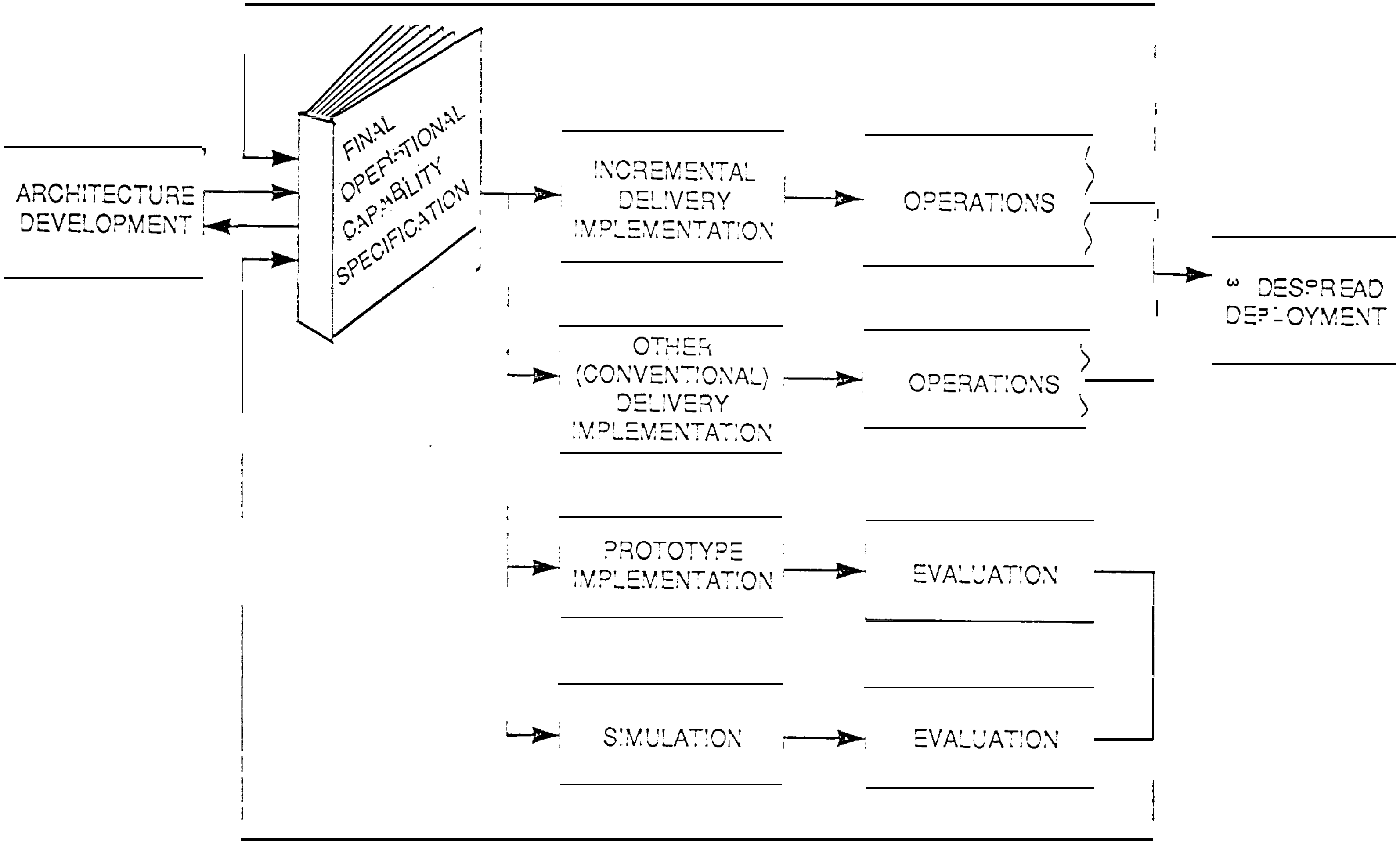

ARCHITECTURE DEVELOPMENT
FINAL OPERATIONAL CAPABILITY SPECIFICATION
INCREMENTAL DELIVERY IMPLEMENTATION
OPERATIONS
OTHER (CONVENTIONAL) DELIVERY IMPLEMENTATION
OPERATIONS
PROTOTYPE IMPLEMENTATION
EVALUATION
SIMULATION
EVALUATION
DESPREAD DEPLOYMENT

Figure 10. Rapid Development of IVHS Systems